

ON THE NUMERICAL SOLUTION OF A FUNCTIONAL DIFFERENTIAL EQUATION PERTAINING TO A WAVE EQUATION

by

David J.N. Wall

Department of Mathematics, University of Canterbury, Christchurch, New Zealand.

No. 57

October, 1990.

Abstract

The numerical solution of the invariant imbedding equation, describing time domain, one dimensional direct scattering from a slab in which the material properties are spatially varying, is considered. It is proven that the equation discretised by the Trapezoidal rule has an asymptotic expansion for the global error involving only even powers of h . This expansion is utilised to generate a high order integration method by use of polynomial extrapolation. The method is suitable for adaptation to parallel computation, and by virtue of this together with its higher order integration, it constitutes a fast algorithm when compared with the current methods of solution of this equation.

AMS(MOS) classifications: 78A45, 65L05, 65R20.

Keywords: Functional differential equation solution, extrapolation method for invariant imbedding equation, asymptotic analysis of discretisation error.

On the Numerical Solution of a Functional Differential Equation Pertaining to a Wave Equation

David J. N. Wall

Dedicated to the memory of Bob Krueger

Abstract. The numerical solution of the invariant imbedding equation, describing time domain, one dimensional direct scattering from a slab in which the material properties are spatially varying, is considered. It is proven that the equation discretised by the Trapezoidal rule has an asymptotic expansion for the global error involving only even powers of h . This expansion is utilised to generate a high order integration method by use of polynomial extrapolation. The method is suitable for adaptation to parallel computation, and by virtue of this together with its higher order integration, it constitutes a fast algorithm when compared with the current methods of solution of this equation.

1. Introduction. The use of a reflection kernel to characterise the scattering of waves in an inhomogeneous medium in the time domain, has been well developed over the last ten years (see for example [4] – [11] and the references cited therein). In [4] it is shown that the use of a reflection kernel combined with invariant imbedding provides a useful and convenient method for the computational solution of a variety of inverse problems. In particular this technique leads to explicit functional equations for the mapping between the reflection kernel, measured at the interface of the inhomogeneous region, and the material functions to be identified. In all previous papers utilising this technique the numerical algorithm used has been the implicit Trapezoidal rule. While this method has provided excellent results it suffers from two major defects in its implementation.

- (i) A low order of approximation of the operator equation.
- (ii) No error estimation.

Our algorithm while still being based on the Trapezoidal rule overcomes both of these problems. In this paper we shall examine an efficient computational method

*Department of Mathematics, University of Canterbury, Christchurch 1, New Zealand.

for solution of the *direct* problem of determination of the reflection kernel, at the interface, when the properties of the inhomogeneous medium are known. Once this kernel is known the reflected wave can be calculated by a convolution.

The method used to solve the problem converts the problem to a functional differential equation which is then solved by the Trapezoidal rule and a global extrapolation technique based on the asymptotic expansion of the global error. This method has several computational advantages not the least of which is that the algorithm is readily adapted to parallel computation. However we should point out the extent to which the algorithm can be parallelised cannot exceed the depth of the extrapolation table (see §3). For a recent review on extrapolation methods see [13]. We shall require several of our results proven in [17] in the sequel.

In the §2 the relevant equations are discussed and it is shown that the problem for determination of the reflection kernel can be reduced to the solution of a functional differential equation. The appropriate properties of the solution of this equation are then displayed. In §4 it is shown that the asymptotic expansion of the solution, obtained via the numerical method described in §3, has even powers of the step size parameter. This enables high order polynomial extrapolation methods to be utilised in the integration of the differential equation. The reflection kernel can be efficiently found when the extrapolation integrator is combined with an adaptive cubic spline interpolation procedure. §5 provides some numerical evaluation of our algorithm.

2. Preliminaries. The one-dimensional spatial wave equation modelling wave motion in the z -direction within a slab, to be investigated in this paper, is

$$U_{zz} - c(z)^{-2}U_{\tau\tau} + \sigma(z)U_{\tau} + \zeta(z)U_z = 0, \quad (2.1)$$

where $c(z)$ is the wave speed of the medium, $\sigma(z)$ is the damping (attenuation) parameter, and ζ is given parameter, again $\zeta = \zeta(z)$. The dependent variable U is dependent upon z and the temporal variable τ and is independent of x, y . The region of interest is $a < z < b$ and as our concern is with the impulse response of the medium we assume σ, ζ are zero outside the interval $[a, b]$ and $c(z) = c(a)$ for $z < a$ and $c(z) = c(b)$ for $z > b$. It is convenient to make the change of coordinates

$$\begin{aligned} x(z) &= \int_a^z c(s)^{-1} ds / \ell, & \ell &= \int_a^b c^{-1}(s) ds, \\ t &= \tau / \ell, & u(x, t) &= U(z, \tau), \end{aligned}$$

where the independent variable x is a travel time coordinate. Equation (2.1) now becomes

$$u_{xx} - u_{tt} + A(x)u_x + B(x)u_t = 0, \quad (2.2)$$

where

$$\begin{aligned} A(x) &= - \frac{d \ln c(z)}{dx} + c\ell\zeta, \\ B(x) &= - c^2\ell\sigma. \end{aligned}$$

We should also make the observation that if the equation (2.1) comes from electromagnetism, and the part of the wave speed which is dependent upon z is the electric

permittivity, the magnetic field equation is of the form (2.1) and then travel time conversion takes it to (2.2) with exactly the same form as the equation obtained via the derivation through the electric field equation (as is done for example in [4]).

Equation (2.1) is sufficiently general to model a variety of electromagnetic and elastic wave scattering phenomena and we shall only consider its normalised version (2.2) in the sequel. The coefficients A and B are thus related to the material parameters and as such $B \leq 0$, with B related to dissipation (energy loss) within the slab. The results presented in this paper however do *not* require the coefficients to have one sign. The coefficients are to have support on the interval $x \in [0, 1]$, and are assumed to be *continuous*. For simplicity in the sequel, as previously stated, we shall assume the slab is matched to the homogeneous exterior region; this will mean

$$\begin{aligned} A(x) &= 0, \quad B(x) = 0, \quad x < 0, \\ A(x) &= 0, \quad B(x) = 0, \quad x > 1. \end{aligned}$$

This requires that in the physical problem the wave speed is continuous in $(-\infty, \infty)$. Means of overcoming this restriction are considered in [10].

In [6] Coronas and Krueger utilise the technique of invariant imbedding to derive from (2.2) the integro-partial differential equation

$$\begin{aligned} R_x^+(x, 1; t) - 2R_t^+(x, 1; t) &= -\frac{1}{2}(A(x) + B(x)) \int_0^t R^+(x, 1; s) R^+(x, 1; t - s) ds \\ &\quad - B(x) R^+(x, 1; t), \quad 0 \leq x \leq 1, \quad 0 \leq t \leq 2(1 - x). \end{aligned} \quad (2.3)$$

This is the imbedding equation describing the reflection kernel at the left-hand interface at location x with the right-hand interface held at $x = 1$. The superscript $+$ is used to signify that this kernel transforms an incident wave moving in the positive x -direction from the left-hand-side of the medium into a reflected wave moving in the negative x -direction. A similar equation holds for the reflection kernel at the right-hand interface, namely $R^-(0, x, t)$, describing the reflection, at location x , of an incident wave from the right-hand medium with the left-hand interface held at $x = 0$. The equation satisfied by R^- can be obtained from (2.3) if the functional dependence on x is replaced by $1 - x$, R_x^+ is then replaced by $-R_x^-$, and the A term is multiplied by -1 since A involves a derivative with respect to x (see [9] for further details).

We define the triangular region in the independent variables (t, x) for which (2.3) is applicable as $D = \{(x, t) \in \mathbb{R}^2 : 0 \leq x \leq 1, 0 \leq t \leq 2(1 - x)\}$. When the problem is one of direct scattering the material functions $A(x)$, $B(x)$, $x \in [0, 1]$, are known and hence

$$R^+(x, 1; 0) = -\frac{1}{4}(A(x) - B(x)), \quad 0 \leq x \leq 1, \quad (2.4)$$

on the side of D where $t = 0$, and it is required to calculate the reflection kernel

$$R^+(0, 1; t), \quad 0 \leq t \leq 2, \quad (2.5)$$

on the side of D where $x = 0$. Observe that we have restricted $t \in [0, 2]$ which is all that is required for one round trip of the incident wave; however $t \in [0, \infty)$

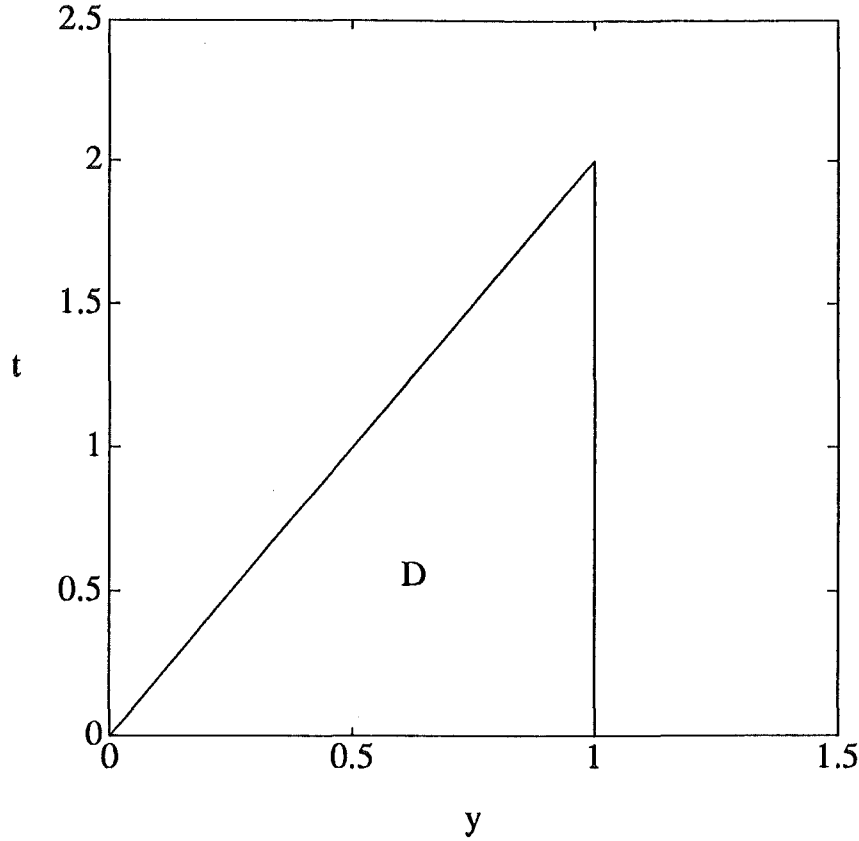


Figure 1. Illustrating the domain of definition of (2.6); the domain for a single return trip is D .

is of interest, and we shall consider this more general setting in a later paper. It should also be noted that the directional derivative on the left-hand side of (2.3) will complicate the direct numerical solution of this equation.

We shall now consider the computational method applied to the solution of the direct scattering problem associated with (2.2) and (2.3). It is essential for the computational scheme by which we solve the problem to convert (2.3) to a functional differential equation. To this end we shall consider the change of independent variable $y = x + t/2$ in order to convert the partial derivatives on the left-hand side of (2.3) into a directional derivative. This will transform the region D into $0 \leq y \leq 1, 0 \leq t \leq 2y$, see Figure 1.

On redefining the dependent variable as $u(t; y) = -2R(y - t/2, t) = -2R(x, t)$ we find

$$\begin{aligned} \frac{du}{dt} = & -\frac{1}{8}(A(y - t/2) + B(y - t/2)) \int_0^t u(s; y)u(t - s; y) ds \\ & + \frac{1}{2}B(y - t/2)u(t; y), \quad 0 \leq t \leq 2y, \quad 0 \leq y \leq 1, \end{aligned} \quad (2.6)$$

with initial conditions

$$u(0; y) = \frac{1}{2}(A(y) - B(y)) = -2R(y, 0), \quad 0 \leq y \leq 1. \quad (2.7)$$

The direct scattering problem can now be stated as: given $u(0; y)$, for $0 \leq y \leq 1$, find $u(t; t/2)$, for $0 \leq t \leq 2$, (or equivalently $u(2y; y)$, for $0 \leq y \leq 1$) from the solution of the Volterra functional differential equation (2.6).

In order to consider the theoretical aspects of the solution of (2.6) we find it convenient to consider equations (2.6), (2.7) through B-space ordinary differential equation theory. We therefore rewrite (2.6) in the standard form

$$\frac{du}{dt}(t; y) - f(t, u(t))(y) = 0, \quad 0 \leq t \leq 2, \quad (2.8)$$

with initial conditions

$$u(0) - u_0 = 0, \text{ and where } u_0 = u(0; y), \quad 0 \leq y \leq 1, \quad (2.9)$$

where $u_0, u \in Y$ and Y is the B-space of continuous functions $C([0, 1])$ with norm

$$\|u\|_Y = \sup_{y \in [0, 1]} \{u(t; y)\},$$

for fixed $t \in [0, 2]$. This will mean for fixed t the function $u(t) = u(t; y) = -2R(x, t)$ will form points of the B-space Y . The mapping function on the right-hand side of (2.8) $f : [0, 2] \times Y \mapsto Y$ is described by

$$f(t, u)(y) = \begin{cases} -\frac{1}{8}(A(y - t/2) + B(y - t/2)) \int_0^t u(s; y)u(t - s; y) ds \\ \quad + \frac{1}{2}B(y - t/2)u(t; y), & 0 \leq t \leq 2y; \\ -\frac{1}{8}(A(0) + B(0)) \int_0^t u(s; y)u(t - s; y) ds \\ \quad + \frac{1}{2}B(0)u(t; y), & 2y \leq t \leq 2. \end{cases} \quad (2.10)$$

Notice f is defined to be continuous at $t = 2y$. We set U to be the space of continuous functions $u : [0 \leq t \leq 2] \mapsto Y$, that is $U = C([0 \leq t \leq 2], Y)$ and the norm for U is

$$\|u\|_U = \sup_{0 \leq t \leq 2} \|u\|_Y, \quad (2.11)$$

then $(U, \|\cdot\|_U)$ is a B-space. We shall also assume that A, B belong to the parameter subspace P , $P = C([0, 1])$ which is a B-space with the appropriate supremum norm. Thus (2.8) describes a system with memory in the t variable, but not in the y variable. This later property will mean (2.8) is particularly efficient for computation. Equation (2.8) is a functional differential equation for u but it cannot be put into the standard form for a Volterra integro-differential equation because of this memory; it is in fact a Volterra functional differential equation.

We define U_α to be the space of continuous functions $U_\alpha = C([0 \leq t \leq \alpha], Y)$, $0 \leq \alpha \leq 2$, with an appropriate norm modelled on (2.11). To proceed further we need the regularity properties of f .

LEMMA 2.1. *With f defined as in (2.10) and with $A, B \in P$ then the mapping f has the following properties.*

(i) $f : [0, 2] \times Y \times P \times P \mapsto Y$.

- (ii) For each $u \in Y$, $f(t, u)$ is continuous with respect to t .
 (iii) For each $t \in [0, 2]$ and $y \in [0, 1]$, f is Fréchet partial differentiable with respect to u and with $f_u = \frac{\partial f}{\partial u}(t, u) : T \mapsto T$ this derivative is defined by the differential

$$(f_u(t, u)v)(y) = \begin{cases} \frac{1}{2}B(y - t/2)v(t; y) - \frac{1}{4}(A(y - t/2) \\ + B(y - t/2)) \int_0^t u(t - s; y)v(s; y) ds, & 0 \leq t \leq 2y; \\ \frac{1}{2}B(0)v(t; y) \\ - \frac{1}{4}(A(0) + B(0)) \int_0^t u(t - s; y)v(s; y) ds, & 2y \leq t \leq 2. \end{cases} \quad (2.12)$$

- (iv) For each $t \in [0, \alpha]$, $0 < \alpha \leq 2$ and $y \in [0, 1]$, f is Lipschitz continuous with respect to u in the ball $B_M = \{u \in U_\alpha : \|u\|_{U_\alpha} \leq M\}$, with Lipschitz constant equal to $\frac{1}{2}\|B\|_P + \frac{1}{4}(\|A\|_P + \|B\|_P)M$.
 (v) For each $t \in [0, 2]$ and $y \in [0, 1]$, f is continuous with respect to A and B .
 (vi) For each $t \in [0, 2]$ and $y \in [0, 1]$, f is Fréchet partial differentiable with respect to A or B .
 (vii) $|f(x, u)| \leq \frac{1}{2}\|B\|_P\|u\|_Y(t) + \frac{1}{8}(\|A + B\|_P) \int_0^t \|u\|_Y(s)\|u\|_Y(t - s) ds$, where the notation $\|u\|_Y(s)$ is used to explicitly illustrate that the scalar quantity $\|u\|_Y$ is a function of the time like variable s .
 (viii) For each $t \in [0, 2]$ and $y \in [0, 1]$, f is infinitely Fréchet partial differentiable with respect to u , with

$$-(f_{uu}(x, u)vw)(y) = \begin{cases} \frac{1}{4}(A(y - t/2 + B(y - t/2)) \\ \times \int_0^t v(t - s; y)w(s; y) ds, & 0 \leq t \leq 2y; \\ \frac{1}{4}(A(0) + B(0)), & 2y \leq t \leq 2, \end{cases} \quad (2.13)$$

and with all derivatives after this one being the zero operator.

- (ix) For each $t \in [0, \alpha]$, $0 < \alpha \leq 2$ and $y \in [0, 1]$, f_u and f_{uu} are Lipschitz continuous, the former with respect to u in the ball B_M , as given in item (iv), and the latter for all U_α . Note that the second derivative of f is a scaled convolution operator.

Proof. Items (i) - (vii) are proven in [17] and (viii), and (ix) follow by straight forward analysis, see [17].

With this result it then follows that (2.8) has only one solution [17].

THEOREM 2.1. *With $u_0 \in Y$, $A, B \in P$, and f possessing the properties of Lemma 2.1 the direct scattering problem (2.8) has exactly one continuously differentiable solution $u \in U$ for all $t \in [0, 2]$. The solution depends continuously on the initial data (2.7) and the parameters A and B .*

COROLLARY 2.1. *With $P \subset C^{(m)}([0, 1])$ and the conditions of Theorem 2.1 holding, (2.8) has exactly one $(m + 1)$ th continuously differentiable solution $u \in U$ for all $t \in [0, 2]$. The solution depends continuously on the initial data (2.7) and the parameters A and B .*

Proof. This follows from Lemma 2.1(vi).

3. Numerical Solution of the Functional Equation. For our convenience in parts of this section we will not show the explicit dependence of u on y . The

numerical scheme we use to solve (2.8) is the implicit Trapezoidal method with the numerical quadrature required in (2.10) also being performed by this method. A uniform step size, h , is chosen with *global* extrapolation being used to obtain a high order method. This uniform step size and the discretisation method chosen has the *important* computational advantage that $u(t)$ for past values of t , $0 \leq t \leq 2$, is available for use in the quadrature rule, for the lag term

$$\int_0^t u(s; y) u(t - s; y) ds,$$

without the necessity of using polynomial interpolation.

The implicitness of the Trapezoidal rule poses no difficulty for (2.8) as by the nature of the non-linearity of u in the integral in (2.10) the resultant algebraic equation for $u(t + h)$ can be solved explicitly. The implicit Trapezoidal method is A-stable, and global extrapolation of this method does not affect this result. However note that if local extrapolation were to be used the higher order methods obtained in the extrapolation table would only be $A(\alpha)$ stable; see for example [16, chapter 6].

We now discuss the numerical difficulties which might be encountered in integrating (2.8). The measure of stiffness of (2.8) is

$$|f_u| < \alpha + 2\beta C,$$

where $\alpha = \frac{1}{2}\|B\|_P$, $\beta = \frac{1}{8}(\|A + B\|_P)$, and

$$C = \exp(2\alpha)\sqrt{\beta w_0} I_1(4\sqrt{\beta w_0}),$$

with I_1 the modified Bessel function of the first kind and $w_0 = \|u_0\|_Y \leq \frac{1}{2}(\|A\|_P + \|B\|_P)$. This follows from Lemma 2.1(iii), (vii) and the comparison equation utilised in the proof of Theorem 2.1 [17]. Typical values of A and B for a particular application are difficult to predict. The possibility of (2.8) being stiff exists for the normal range of parameters is highly problem dependent, for example $|B|$ can be of the order of 10^6 . One of the major problems associated with the Trapezoidal rule is that although the method is A-stable, and therefore numerical instability is not a problem, it is not L-stable [14, Chapter 8]. This will mean the method will not provide an accurate solution unless the initial transient response is integrated accurately, that is with small h values. However as the initial transient is all important in inverse problems one presumes that this transient will be calculated accurately. Then the error in this rapidly decaying term will be small and the Trapezoidal method will yield an accurate answer with moderate values of h for the rest of the integration.

The integral term in (2.10) is of convolutional form and straight-forward evaluation of this integral at t , where $h = t/n$, will have a computational cost of $O(n^2)$ flops. At first sight it might seem possible to utilise the Fast Fourier Transform (FFT) algorithm to reduce this cost as is done in [12]. In [12] the FFT is applied in a novel manner to a Volterra integral equation with a convolution kernel. The important difference between this equation and (2.8) is that in the Volterra integral

equation the kernel in the integral is *known* for all values of the independent variable, whereas this is not the case in (2.8). A moment's reflection shows that this FFT 'trick' is not possible for a convolution Riccati equation such as (2.8). However the symmetry of the quadratic integrand in the integral operator in (2.10) and in its Trapezoidal summation, namely

$$\int_0^t u(t-s)u(s) ds = h \sum_{j=0}^n {}'' u_j u_{n-j}, \quad h = t/n, \quad (3.1)$$

indicates that the computational cost can be halved to $O(n^2/2)$. In (3.1) the double prime on the summation signifies that the first and last term are to be halved.

3.1 Global Error Estimation of the solution. Consider the discrete method just discussed and denote its numerical solution by $u(t, h) = U_n$, to explicitly show its dependence on step-size. Then it is shown in Theorem 4.1 that the global error has an asymptotic expansion of the form

$$u(t) - u(t, h) = e_2 h^2 + e_4 h^4 + \dots + e_{2J} h^{2J} + O(h^{2J+2}), \quad (3.2)$$

with the $\{e_j\}_{j=1}^{2J}$ being independent of h . If the numerical solution at t when computed for the different step sizes $h, h/2, h/4, \dots$ is denoted by $T_{i,0} = u(t, h/2^i)$, then the extrapolation tableau

$$\begin{array}{ccc} & T_{0,0} & \\ & T_{1,0} & T_{1,1} \\ T_{2,0} & T_{2,1} & T_{2,2} \end{array}$$

can be calculated according to the Neville-Aitken algorithm

$$T_{i,k} = T_{i,k-1} + \frac{T_{i,k-1} - T_{i-1,k-1}}{4^k - 1}, \quad i \geq k \geq 1.$$

This algorithm cancels the leading term in the asymptotic expansion of $T_{i,k-1}$ so that the global error of $T_{i,k}$ is $O(h^{2k+2})$, that is

$$u(t) - T_{i,k} = \gamma_{ik} e_{2k} h^{2k+2} + O(h^{2k+4}).$$

Thus by building up the tableau, if we are prepared to solve the problem for $i+1$ step sizes we can generate an estimate for $u(t)$ that is at least asymptotically $O(h^{2i+2})$ accurate.

We now discuss the convergence test. Define the row difference

$$d_r(i, k) = T(i, k) - T(i, k-1),$$

then for a prescribed tolerance *tol* the extrapolation tableau is calculated until row convergence in two successive rows is obtained; that is

$$|d_r(i-1, k)| \leq \text{tol} \times |T(i-1, k)|$$

and

$$|d_r(i, k)| \leq tol \times |T(i, k)|.$$

By these considerations the d_r estimate the relative error of $T_{i,k}$, and then the more accurate value $T_{i,k}$ is to be accepted as a numerical approximation to $u(t)$.

If a column difference

$$d_c(i, k) = T(i, k) - T(i - 1, k),$$

is defined we observe from (3.2)

$$r(i, k - 1) = \frac{d_c(i - 1, k - 1)}{d_c(i, k - 1)} \rightarrow 4^k, \quad K > 1, \quad \text{as } i \text{ increases.} \quad (3.3)$$

Consequently this ratio can be checked numerically. However this number will lose significance rapidly on a finite precision machine.

We again note this algorithm lends itself readily to parallel computation in that (2.8) can be simultaneously integrated out to $2y$ for different step sizes and then global extrapolation carried out.

3.2 Adaptive Interpolation. It remains to describe how the remainder of the problem is solved. The differential equation must be integrated to $0 \leq t \leq 2y$, for all $0 \leq y \leq 1$, to obtain $u(2y, y)$. This is achieved with minimum cost by use of an adaptive interpolation algorithm which will minimise the number of points y at which (2.8) will have to be integrated. We will interpolate $u(2y, y)$ for $0 \leq y \leq 1$ by a cubic spline. In order to provide a foundation for the adaptive algorithm it is necessary to estimate the error bound between the cubic interpolatory spline and $u(2y, y)$. This will ensure that a suitable sub-division of $y \in [0, 1]$ is made which is dependent upon the magnitude of the fourth derivative of $u(2y, y)$. A decision as to which interpolatory cubic spline is to be used must be made with possible candidates being from the 'derivative free' class. Therefore possible candidates would be the *not-a-knot* or the *clamped spline obtained by forcing the end derivatives of the spline to agree with that of a local cubic interpolant*. The error for these cubic 'derivative free' interpolating splines on the mesh $[0 = y_0, y_1, \dots, y_n = 1]$ is given by [1]

$$|e_S(y)| = |S(y) - u(y)| \leq u^{(4)}(2\xi, \xi) h_{max}^4 C_S, \quad \xi \in (0, 1), \quad (3.4)$$

where C_S is an appropriate error constant, and h_{max} is the maximum mesh interval used on $y \in [0, 1]$. Numerical experiments indicate that with *uniform meshes* the not-a-knot spline $|C_S| < 0.035$ and for the other $|C_S| < 0.040$, [2]. The strategy of our procedure is similar to the one taken in adaptive quadrature. A crude uniform sub-division of $[0, 1]$ is performed, with mesh size h , and by taking the first four nodes a local cubic interpolation polynomial is fitted. This polynomial is then evaluated at the three mid-points between the four nodes. The functional differential equation is also integrated at these points and the absolute maximum error of the polynomial can then be estimated. The assumption here is that the fourth derivative of $u(2y, y)$ is almost constant over the interval being examined. If this error is *not less* than the constant e_L , to be described shortly, then another cubic interpolation polynomial is

fitted to the first four nodes formed when the interval size is now $h/2$. The accuracy test is then repeated and the process continued until the accuracy test is achieved. The algorithm thereby *marches* towards the node $y = 1$ so ensuring each local cubic interpolation polynomial on each successive four nodes of the resultant mesh has an error less than e_L . Recall that the error between the local cubic interpolation polynomial and u , with a uniform spacing h , is given by

$$|e_3(y)| = |p_3(y) - u(y)| \leq \frac{u^{(4)}(2\xi, \xi)}{4!}(h)^4 C_3, \quad \xi \in (y_i, y_{i+3}), \quad 0 \leq i \leq n-3$$

where $C_3 = 1/16$. The error when using the cubic spline on this interval is given by (3.4), but with $h_{max} = h$. Then if we require the global error for the interpolatory spline to be less than tol , a prescribed tolerance, that is $|e_S| < tol$, it is seen that when using the local cubic strategy we must ensure that the error incurred by this polynomial is

$$|e_3| < e_l = \frac{C_3}{C_S} \times tol.$$

The error tolerance for the spline fit will then be satisfied. Observe that although this is only a local result the global error is also bounded by tol .

It should be noted that other adaptive approximation algorithms for cubic spline interpolation have been suggested [15, §21.3], [8, Chapters XII and XI]. However these references solve this fully non-linear problem by iterative techniques and do not use the local cubic approximation ideas incorporated here.

4. Asymptotic Error Expansion of the Discretisation Method. Prior to proving the main theorem in this section we shall define our terminology. By following the notation of [16] it is convenient to define the Banach spaces,

$$\begin{aligned} E &= C^{(1)}([0 \leq t \leq 2], Y), \\ E^o &= Y \times C([0 \leq t \leq 2], Y), \end{aligned}$$

with norm in E^o

$$\left\| \begin{array}{c} w_0(y) \\ w(t)(y) \end{array} \right\|_{E^o} = \|w_0\|_Y + \|w(t)\|_U.$$

Observe that the superscript 1 on $C^{(1)}$ denotes that it is the space of continuously differentiable functions with respect to t . For this section we define the operator F to stand for (2.8) and (2.9), so that

$$Fu = \left[\begin{array}{c} (u(0) - u_0)(y) \\ u^{(1)}(t; y) - f(t, u(t))(y) \end{array} \right] \in E^o, \quad u \in E,$$

with the notation that $u^{(1)}$ denotes the derivative du/dt . The discretisation of F is associated with the finite dimensional Banach spaces E_n, E_n^o for the discretisation parameter n . Note these spaces are different for each value of n . We assume n is chosen from a set $\mathbb{N}' \subset \mathbb{N}$, \mathbb{N} denoting the natural numbers. The discretised operator is denoted by F_n with $F_n : E_n \mapsto E_n^o$, and appropriate restriction operators are denoted by r_n, r_n^o , with $r_n : E \mapsto E_n, r_n^o : E^o \mapsto E_n^o$. We note it is possible

to simply quote Stetter's [16, Theorem 1.3.1] and then to show that our problem satisfies all the conditions required to complete this. However his theorem requires several technical definitions and for our problem it is simpler, and more transparent, to prove the result directly. First recall the definition of the consistency error map.

DEFINITION 4.1. The sequence of mappings, dependent upon n , $\Lambda_n : E \mapsto E^o$, $n \in \mathbb{N}'$, such that

$$F_n r_n w = r_n^o [F + \Lambda_n] w, \quad \text{for all } w \text{ in the domain of } F,$$

is called the consistency error map.

Remark 4.1. The consistency error $\mathcal{L}_n = F_n r_n w - r_n^o F w$ is generally defined for $n \rightarrow \infty$, whereas Definition 4.1 is taken to apply for all $n \in \mathbb{N}'$.

The commutativity diagram illustrates the relationship between the various spaces and definitions.

$$\begin{array}{ccc} E & \xrightarrow{F+\Lambda_n} & E^o \\ \downarrow r_n & & \downarrow r_n^o \\ E_n & \xrightarrow{F_n} & E_n^o \end{array}$$

With ϵ_n , U_n , and u denoting the global error, the exact solution of the discretised problem and the exact solution of the continuous problem respectively, the definition of the global error implies that these terms are related through

$$\epsilon_n = U_n - r_n u \in E_n, \quad n \in \mathbb{N}', \quad (4.1)$$

So applying the operator F_n to (4.1) it follows

$$F_n(\epsilon_n + r_n u) = F_n U_n = 0. \quad (4.2)$$

To find a general expression for the global error observe from (4.1)

$$\epsilon_n = -F_n^{-1} F_n r_n u + F_n^{-1} 0,$$

as $F_n U_n = 0$ and further

$$\epsilon_n = -F_n^{-1} \mathcal{L}_n + F_n^{-1} 0, \quad \text{as } n \rightarrow \infty,$$

where \mathcal{L}_n is the consistency error of the discretised problem to the continuous one. Therefore if F_n^{-1} satisfies a Lipschitz condition uniformly in n then

$$\|\epsilon_n\| = \|F_n^{-1}\| \|\mathcal{L}_n\|. \quad (4.3)$$

Equation (4.3) seems to imply that it is necessary find the inverse to F_n to obtain the form of the global error. However Stetter [16] has devised a method to circumvent this difficulty and go directly to the asymptotic expansion of the global error. We shall follow his method here.

If

$$F(w + \sum_{j=1}^J h^j e_j) + \Lambda_n(w + \sum_{j=1}^J h^j e_j) = O(h^{J+1}), \quad \text{as } h \rightarrow 0, \quad (4.4)$$

where the e_j are to be defined, then Definition 4.1 shows

$$F_n(r_n(w + \sum_{j=1}^J h^j e_j)) = O(h^{J+1}).$$

Equations (4.2) and (4.4) then imply we have an asymptotic expansion of the global error

$$\epsilon_n = r_n[\sum_{j=1}^J h^j e_j] + O(h^{J+1}) \quad \text{as } h \rightarrow 0, \quad (4.5)$$

where e_j are now the elements of the asymptotic global discretisation error. In (4.5) if the discretisation scheme is convergent of order p then the $e_j = 0$, $1 \leq j \leq p-1$, and also if the global discretisation contains only even powers of h , then $e_{(2j-1)} = 0$, $1 \leq j \leq [J/2]$. We may now state the central result of this section.

THEOREM 4.1. *The discretisation of (2.8) by the Trapezoidal rule implies that the solution of the discretised equation will possess a unique asymptotic expansion to order $2J$ of the form*

$$\epsilon_n = r_n(\sum_{j=1}^J h^{2j} e_{2j}) + O(h^{2J+2})$$

provided the conditions of Theorem 2.1 and Corollary 2.1, with $m = 2J + 1$, are satisfied.

Proof. It is obvious from the definition of F , and its discretisation, that the first component of Λ_n is zero. The second component of the consistency error mapping is given by

$$\begin{aligned} & \frac{w(t + h/2) - w(t - h/2)}{h} - \frac{1}{2}[f(t + h/2, w(t + h/2)) + f(t - h/2, w(t - h/2))] \\ & - w'(t) - f(t, w(t)), \quad w \in E, \end{aligned} \quad (4.6)$$

for notational simplicity we have not explicitly written the Trapezoidal quadrature rule within the square brackets, but the understanding is that each integral term within this bracket is to be replaced by its Trapezoidal approximation, see (3.1). Use of Taylor's formula and the Euler-MacLaurin formula in (4.6) then verifies for each positive integer J , the Λ_n , $n \in \mathbb{N}'$, possesses an asymptotic expansion to order $2J$ and its second component is

$$\begin{aligned} & \sum_{j=1}^J \left[\frac{1}{(2j+1)!} D_t^{(2j+1)} w(t) - \frac{1}{(2j)!} D_t^{(2j)} f(t, w(t)) \right. \\ & \left. + \frac{B_{2j}}{(2j)!} \left(\sum_{l=1}^{J-2l} \left(\frac{h}{2} \right)^{2l} D_t^{(2l)} \left\{ g(t) \frac{d^{(2l-1)}}{ds^{(2l-1)}} (w(t-s)w(s)) \right\} \Big|_{s=t} \right) \right] \left(\frac{h}{2} \right)^{2j}, \end{aligned} \quad (4.7)$$

where $g(t) = -\frac{1}{8}(A(y - t/2) + B(y - t/2))$ and the B_{2j} denote the even Bernoulli numbers. The smoothness assumption on w to achieve (4.7) is that $w \in C^{2J+3}(0 \leq t \leq 2)$. We note if $u \in C^{2J+3}(0 \leq t \leq 2)$ equation (4.7) is the consistency error of the problem. It should be observed that in general unless the solution is smooth enough, the actual consistency error obtained, and the consistency error mapping need not be the same. Equation (4.7) can also be written

$$\|r_n^\circ[\Lambda_n w - \sum_{j=1}^J h^{2j} \lambda_{2j} w]\|_{E_n^\circ} = O(h^{2J+2}), \quad \text{as } h \rightarrow 0, \quad (4.8)$$

where the operators λ_{2j} (which are independent of h) are given by the coefficient of h^{2j} in (4.7). Notice that the first component of this mapping which is associated with the initial condition is just the zero operator. The asymptotic expansion for the global error is expected to contain only even powers of h because of (4.7). Throughout the remainder of the proof the superscript (k) appended to operators will denote the Fréchet derivatives of the operators with respect to the variable w . It is important for the sequel to note that the implicit Trapezoidal method introduced in §3 is stable, that is F_n^{-1} satisfies a Lipschitz condition, compare with (4.3), and also $(F^{(1)}(w))^{-1}$ exists.

The proof can now proceed by showing that (4.4) holds when $w = u$, and u is the solution of $Fu = 0$. To do this the mappings F and Λ_n are expanded by Taylor series. First observe that:

(i) The λ_{2j} are non-linear operators because of the second term in (4.7) and $\lambda_{2j}^{(1)}$ is given by λ_{2j} , but the second term involving f is replaced by f_w , also

$$\lambda_{2j}^{(2)} = \begin{bmatrix} 0 \\ \frac{-1}{(2j)!} D_t^{(2j)} f_{ww}(t, w(t)) e^2(t) \end{bmatrix},$$

and $\lambda_{2j}^{(m)} = 0$, $m > 2$.

(ii) The Fréchet derivatives of F are

$$F^{(1)}(w)e = \begin{bmatrix} e(0) \\ e'(t) - f_w(t, w(t))e(t) \end{bmatrix},$$

$$F^{(2)}(w)e^2 = \begin{bmatrix} 0 \\ f_{ww}(t, w(t))e^2(t) \end{bmatrix},$$

with the linear operators f_w and f_{ww} given by Lemma 2.1, and with e' denoting de/dt .

Now $F(w + \sum_{j=1}^J h^{2j} e_{2j}) + \Lambda_n(w + \sum_{j=1}^J h^{2j} e_{2j})$ expands to

$$\begin{aligned} F(w) + F^{(1)}(w) \left(\sum_{j=1}^J h^{2j} e_{2j} \right) + \sum_{j=1}^J h^{2j} \lambda_{2j} w + \sum_{j=1}^J h^{2j} \lambda_{2j}^{(1)}(w) \left(\sum_{k=1}^{J-j} h^{2k} e_{2k} \right) \\ + \frac{1}{2!} [F^{(2)}(w) + \lambda_{2j}^{(1)}(w)] \left(\sum_{j=1}^J h^{2j} e_{2j} \right)^2 + R_{2J}(w; e_2, e_4, \dots, e_{2J}), \end{aligned} \quad (4.9)$$

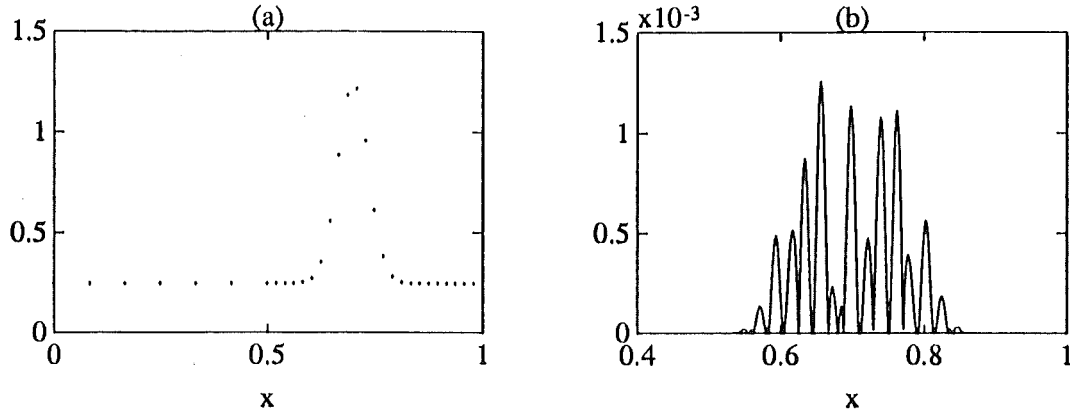


Figure 2. An example illustrating the distribution of nodes (a) and the pointwise absolute error (b) obtained from the adaptive interpolation algorithm when fitting a ‘bell-shaped’ function $0.25 + \exp[-400(x-0.7)^2]$; the dots indicate the interpolation nodes necessary to attempt to achieve a global error $tol < 10^{-2}$.

and when $w \equiv u$ the first term disappears and our previous assumptions ensure

$$\|R_{2J}(w; e_2, e_4, \dots, e_{2J})\| = O(h^{2J+2}),$$

if the e_{2j} , $1 \leq j \leq J$ are defined from

$$\begin{aligned} e_{2j}(0) &= 0, \\ e'_{2j}(t) - f_u(t, u(t))e_{2j}(t) &= b_{2j}, \quad t \in [0, 2], \end{aligned} \tag{4.10}$$

with

$$\begin{aligned} b_2 &= -\lambda_2 u(t) \\ b_4 &= -\lambda_4 u(t) - \lambda_2 e_2(t) - \frac{1}{2} f_{uu}(t, u) e_2^2 \\ b_6 &= -\lambda_6 u(t) - \lambda_4 e_4(t) - f_{uu}(t, u) e_2 e_4 \\ &\text{etc.} \end{aligned}$$

Observe the b_j are the coefficients of h^{2j} on the in (4.9). Note that with $u \in C^{(2J+3)}([0 \leq t \leq 2], Y)$ and f_u, f_{uu} satisfying Lipschitz conditions (Lemma 2.1(viii)) it follows that the recursive definition of the e_{2j} is well defined from (4.10). We observe that (4.10) have been obtained by equating like powers of h of the second and third of the terms in (4.9) to those of the fourth term. This ensures (4.10) reduces to $O(h^{2J+2})$ and our result follows from (4.4) and (4.5).

Remark 4.2. The result of Theorem 4.1 is only true if the implicit trapezoidal rule is solved exactly at each step; this will mean the implicitness is removed either analytically or by solving the equation by a recursive method to convergence. That this is possible for (2.8) is shown in §3.

5. Numerical Results. We illustrate results of the algorithm presented in §3 in Figures 2-4 and Tables 1 and 2.

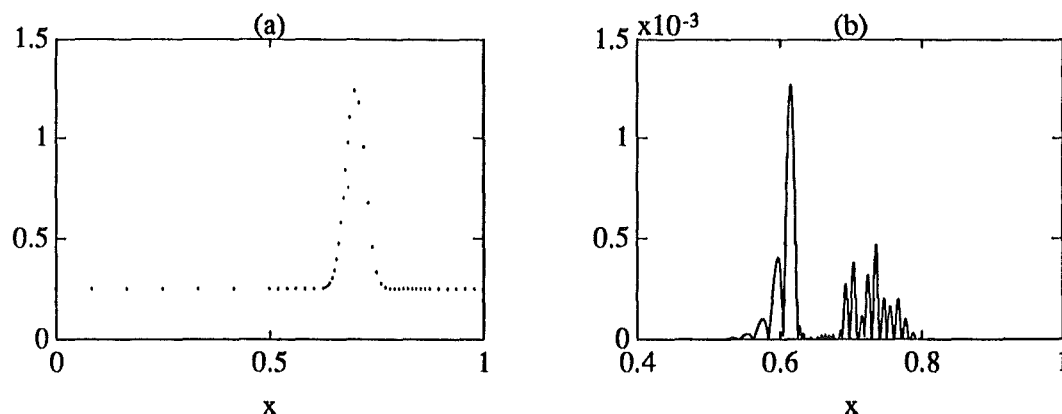


Figure 3. An example illustrating the distribution of nodes (a) and the pointwise absolute error (b) obtained from the adaptive interpolation algorithm when fitting a 'bell-shaped' function $0.25 + \exp[-1000(x - .7)^2]$; the dots indicate the interpolation nodes necessary to attempt to achieve a global error $tol < 10^{-3}$.

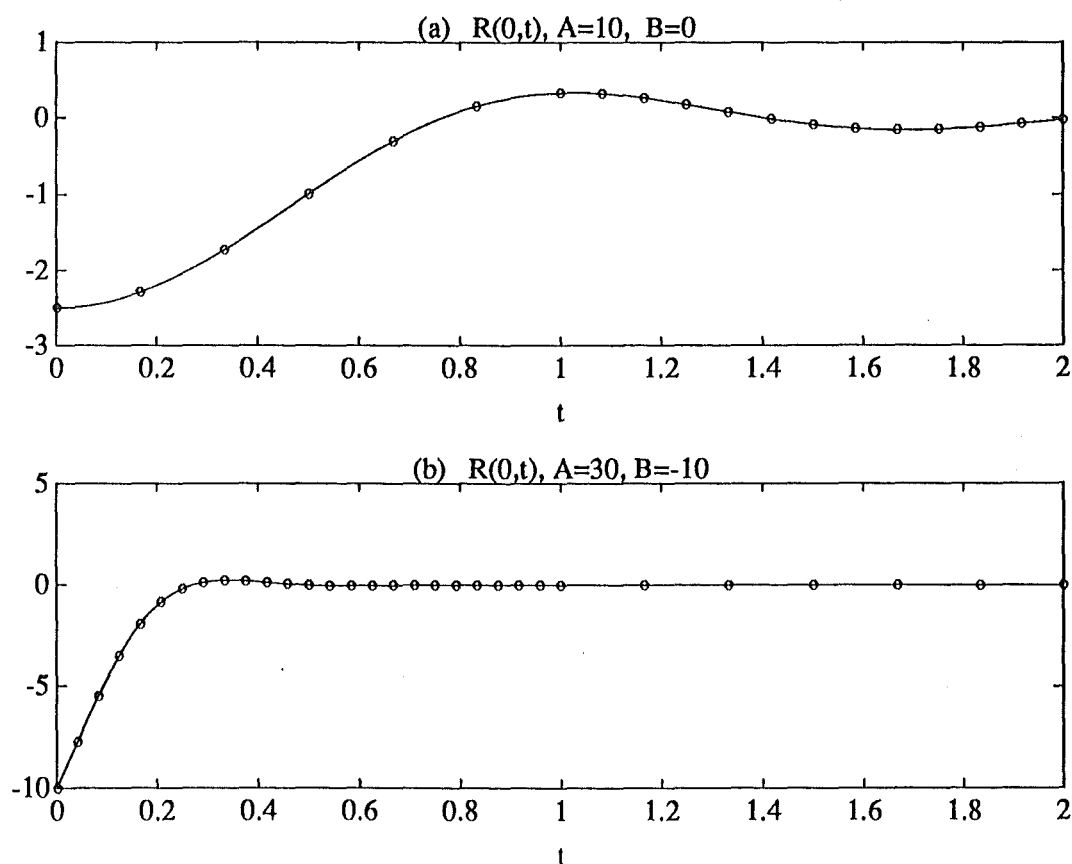


Figure 4. An example illustrating the distribution of nodes obtained from the adaptive integration algorithm when $tol = 10^{-4}$ and (a) $A = 10, B = 0$, (b) $A = 30, B = -10$.

i	$T(i, 0)$	$T(i, 1)$	$T(i, 2)$	$T(i, 3)$	$T(i, 4)$
0	1.12209789152				
1	1.02824308029	0.996958143217			
2	1.00284087142	0.994373468466	0.994201156816		
3	0.996360203086	0.994199980307	0.994188414430	0.994188212170	
4	0.994731758755	0.994188943979	0.994188208223	0.994188204950	0.994188204922

i	$r(i, 0)$	$r(i, 1)$	$r(i, 2)$
1	3.7		
2	3.9	14.9	
3	4.0	15.7	61.8

Table 1. Extrapolation table for $R(0, 0.5)$, when $A = 10$ and $B = 0$, and ratios of extrapolation columns. The exact solution for this case is $R(0, .25) = 0.99418820492855$. (See exact solution in [17, equation 4.8]. Note there is a typographical error in this equation, the left-hand side should be divided by (2β)).

i	$T(i, 0)$	$T(i, 1)$	$T(i, 2)$	$T(i, 3)$	$T(i, 4)$
0	0.461940032555				
1	0.310108818451	0.259498413750			
2	0.247628259314	0.22680140626	0.224621605770		
3	0.231254187457	0.225796163504	0.225729147320	0.225746727344	
4	0.227133359773	0.225759750545	0.225757323014	0.225757770247	0.225757813553

i	$r(i, 0)$	$r(i, 1)$	$r(i, 2)$
1	2.4		
2	3.8	32.5	
3	4.0	27.6	39.3

Table 2. Extrapolation table for $R(0, 0.35)$, when $A = 30$ and $B = -10$, and ratios of extrapolation columns. The exact solution for this case $R(0, 0.35) = 0.22575780953328$. If the extrapolation is continued further it is found that $T(6, 4)$ has converged to at least 12 significant decimal digits; these extra columns have not been shown for reasons of space.

The diagrams in Fig. 2 and 3 illustrate the nodes chosen by the adaptive interpolation routine when the required global error is specified to be the values shown in the figure caption. It is obvious from the diagrams that more points are taken in regions where the fourth derivative is largest. The diagrams also show in practice that the actual error obtained, which has been found by computation, can be lower than the error bound estimates of §3.1. However as Fig. 3(b) illustrates if the rate of change in the function is too great the algorithm can be *fooled* and the required accuracy not achieved, this can be corrected by asking for a higher

accuracy than is actually required. Figures 4(a) and 4(b) show the result of using the full algorithm to solve (3.1) for the accuracies and with parameters specified in the figure caption. The nodes and function values to be utilised in the spline are illustrated by circles showing the coarser mesh which suffices when the fourth derivative of the reflection kernel is small. Note that when the spline is fitted through these values a smooth C^2 function shown by the full line is obtained.

Tables 1 and 2 show the extrapolation tableaux for the integration algorithm and they illustrate convergence, to seven significant decimal places, in the integration rule with consistency errors $O(h^8)$ and $O(h^{10})$, respectively. Listed below the T 's are the ratios for the respective tableaux. Observe for Table 1 the basic Trapezoidal rule $T(4,0)$ achieves 3 significant decimal digits of accuracy whereas the rules $T(2,2)$ and $T(4,4)$ have 4 and 11 significant decimal digits of accuracy respectively. The Trapezoidal rule would require very many more intervals to achieve the accuracy of $T(4,4)$. In Table 2 similar high accuracy results are obtained by the extrapolation integrator.

It should be observed that the numerical values in the ratio tables in many cases have not achieved the asymptotic value predicted by (3.3); this is because the ratios shown are only for low values of i .

6. Conclusions. Theorem 4.1 has provided a theoretical basis for a high order extrapolation algorithm that still provides all the advantages of the simple Trapezoidal rule, namely convenient integration of the lag term coupled with stability properties. The transformation employed in §2 enables application of an efficient adaptive interpolation algorithm. Together these two algorithms provide efficient solution of the functional differential equation of §2. Although the transformation is not possible in problems involving multiple wave speeds it is possible to produce adaptive high-order integration algorithms for such problems, and this is being currently investigated.

Acknowledgement.

I wish to thank my colleague Bob Broughton for discussions on the numerical solution of ordinary differential equations.

REFERENCES

- [1] R.K. Beatson, *On convergence of cubic spline interpolation schemes*, SIAM J. Numer. Anal., 23, (1986), pp. 903-912.
- [2] R.K. Beatson and E. Chacko, *Which cubic spline should one use?*, Department of Mathematics Preprint, University of Canterbury, Christchurch, New Zealand, 1990.
- [3] H. Brunner and P.J. van der Houwen, *The Numerical Solution of Volterra Equations*, Elsevier Science, Amsterdam, 1986.
- [4] J.P. Coronas, M.E. Davison and R.J. Krueger, *Wave splittings, invariant imbedding and inverse scattering*, in Inverse Optics, Proc. SPIE 413, (A.J. Devaney, Ed.), pp. 102-106 SPIE, Bellingham, Wash., 1983.
- [5] ———, *The effects of dissipation in one-dimensional inverse problems*, in Inverse Optics, Proc. SPIE 413, (A.J. Devaney, Ed.), pp. 107-114 SPIE,

- Bellingham, Wash., 1983.
- [6] J.P. Coroné and R.J. Krueger, *Obtaining scattering kernels using invariant imbedding*, J. Math. Anal. Appl., 95, (1983), pp. 393-415.
 - [7] J.P. Coroné, R.J. Krueger, and C.R. Vogel, *The effects of noise and band-limiting on a one dimensional time dependent inverse scattering technique*, in *Review of Progress in Quantitative Nondestructive Evaluation Vol 4*, (D.O. Thompson and D.E. Chimenti, Ed.) pp. 551-558 Plenum New York, 1985.
 - [8] C. de Boor, *A Practical Guide to Splines*, Springer-Verlag, New York, 1978.
 - [9] G. Kristensson and R.J. Krueger, *Direct and inverse scattering in the time domain for a dissipative wave equation. I. Scattering operators*, J. Math. Phys., 27, (1986), pp. 1667-1682.
 - [10] ———, *Direct and inverse scattering in the time domain for a dissipative wave equation. III. Scattering operators in the presence of a phase velocity mismatch*, J. Math. Phys., 28, (1987), pp. 360-370.
 - [11] ———, *Direct and inverse scattering in the time domain for a dissipative wave equation. IV. Use of phase velocity mismatches to simplify inversions*, Inverse Problems, 5, (1989), pp. 375-388.
 - [12] E. Hairer, CH. Lubich, & M. Schlichte, *Fast numerical solution of non-linear Volterra convolution equations*, SIAM J. Sci. Stat. Comput., 6, (1985), pp. 532-541.
 - [13] P. Deuffhard, *Recent progress in extrapolation methods for ordinary differential equations*, SIAM Review, 27, (1985), pp. 505-535.
 - [14] J.D. Lambert, *Computation Methods in Ordinary Differential Equations*, John Wiley, New York, 1973.
 - [15] M.J.D. Powell, *Approximation Theory and Methods*, Cambridge University Press, Cambridge, 1981.
 - [16] H.J. Stetter, *Analysis of Discretisation Methods for Ordinary Differential Equations*, Springer-Verlag, New York, 1973.
 - [17] D.J.N. Wall, *On some Differential Equations arising in a Time Domain Inverse Scattering Problem for a Dissipative Wave Equation*, J. Transport Theory and Statistical Physics, (1991), to appear.